Submitted to the Annals of Applied Statistics

Preprint ANL/MCS-P2091-0512

STOCHASTIC APPROXIMATION OF SCORE FUNCTIONS FOR GAUSSIAN PROCESSES

By Michael L. Stein[*], Jie Chen[†] and Mihai Anitescu[†]

University of Chicago[*] *and Argonne National Laboratory*[†]

We discuss the statistical properties of a recently introduced unbiased stochastic approximation to the score equations for maximum likelihood calculation for Gaussian processes. Under certain conditions, including bounded condition number of the covariance matrix, the approach achieves $O(n)$ storage and nearly $O(n)$ computational effort per optimization step, where n is the number of data sites. Here, we prove that if the condition number of the covariance matrix is bounded, then the approximate score equations are nearly optimal in a well-defined sense. Therefore not only is the approximation efficient to compute, but it also has comparable statistical properties to the exact maximum likelihood estimates. We discuss a modification of the stochastic approximation in which design elements of the stochastic terms mimic patterns from a 2^n factorial design. We prove these designs are always at least as good as the unstructured design, and we demonstrate through simulation that they can produce a substantial improvement over random designs. Our findings are validated by numerical experiments on up to 1 million data sites that include fitting of numerical output from a problem in geodynamics.

1. Introduction.

Gaussian process models are widely used in spatial statistics and machine learning. In most applications, the covariance structure of the process is at least partially unknown and must be estimated from the available data. Likelihood-based methods, including Bayesian methods, are natural choices for carrying out the inferences on the unknown covariance structure. For large datasets, however, calculating the likelihood function exactly may be difficult or impossible in many cases.

Assuming we are willing to specify the covariance structure up to some parameter $\theta \in \Theta \subset \mathbb{R}^p$, the generic problem we are faced with is computing the loglikelihood for $Z \sim N(0, K(\theta))$ for some random vector $Z \in \mathbb{R}^n$ and K an $n \times n$ positive definite matrix indexed by the unknown θ. In many applications, there would be a mean vector that also depends on unknown parameters; but since unknown mean parameters generally cause fewer computational difficulties, for simplicity we will assume the mean is known to be

AMS 2000 subject classifications: Primary 60G15; secondary 65C60

Keywords and phrases: Gaussian process, unbiased estimating equations, Hutchinson trace estimators, maximum likelihood, iterative methods, preconditioning

0 throughout this work. The simulations in Section 5 and the application in Section 6 all first preprocess the data by taking a discrete Laplacian, which would filter out any mean function that was linear in the coordinates, so that the results in those sections would be unchanged for such mean functions. The loglikelihood is then, up to an additive constant, given by

$$\mathcal{L}(\theta) = -\frac{1}{2} Z' K(\theta)^{-1} Z - \frac{1}{2} \log\det\{K(\theta)\}.$$

If K has no exploitable structure, the standard direct way of calculating $\mathcal{L}(\theta)$ is to compute the Cholesky decompositon of $K(\theta)$, which then allows $Z'K(\theta)^{-1}Z$ and $\log\det\{K(\theta)\}$ to be computed quickly. However, the Cholesky decomposition generally requires $O(n^2)$ storage and $O(n^3)$ computations, either of which can be prohibitive for sufficiently large n. Iterative methods often provide an efficient (in terms of both storage and computation) way of computing $K(\theta)^{-1}Z$ and are based on being able to multiply arbitrary vectors by $K(\theta)$ rapidly. Therefore, computing $\log\det\{K(\theta)\}$ is commonly the major obstacle in carrying out likelihood calculations.

If our goal is just to find maximum likelihood estimates (MLEs) and the corresponding Fisher information matrices, we may be able to avoid the computation of the log determinants by considering the score equations, which are obtained by setting the gradient of the loglikelihood equal to 0. Specifically, defining $K_i = \frac{\partial}{\partial\theta_i} K(\theta)$, the score equations for θ are given by (suppressing the dependence of K on θ)

$$\frac{1}{2} Z' K^{-1} K_i K^{-1} Z - \frac{1}{2} \mathrm{tr}(K^{-1} K_i) = 0 \tag{1.1}$$

for $i = 1, \ldots, p$. If these equations have a unique solution for $\theta \in \Theta$, this solution will generally be the MLE. Computing the first term requires only one solve in K, but the second term requires n solves and hence may not be any easier to compute than the log determinant.

Recently Anitescu, Chen and Wang (2012) analyzed and demonstrated a stochastic approximation of the second term based on the Hutchinson trace estimator (Hutchinson, 1990). To define it, let $U_1, \ldots, U_N$ be iid random vectors in $\mathbb{R}^n$ with iid symmetric Bernoulli components; that is, taking on values 1 and -1 each with probability $\frac{1}{2}$. Define a set of estimating equations for θ by

$$g_i(\theta, N) = \frac{1}{2} Z' K^{-1} K_i K^{-1} Z - \frac{1}{2N} \sum_{j=1}^{N} U_j' K^{-1} K_i U_j = 0 \tag{1.2}$$

for $i = 1, \ldots, p$. Throughout this work, E_θ means to take expectations over $Z \sim N(0, K(\theta))$ and over the U_j's as well. Since $E_\theta(U_1' K^{-1} K_i U_1) =$

$\mathrm{tr}(K^{-1}K_i)$, $E_\theta g_i(\theta, N) = 0$ and (1.2) provides a set of unbiased estimating equations for θ. Therefore, we may hope that a solution to (1.2) will provide a good approximation to the MLEs. The unbiasedness of the estimating equations (1.2) requires only that the components of the U_j's have mean 0 and variance 1; but, subject to this constraint, Hutchinson (1990) shows that, assuming the components of the U_j's are independent, taking them to be symmetric Bernoulli minimizes the variance of $U_1'MU_1$ for any $n \times n$ matrix M.

In addition to reducing computations relative to computing the exact score function, the approximate score function has at least two other attractive features. First, assuming the solves in K are obtained by using iterative methods and that the elements of K can be computed as needed, the algorithm uses only $O(n)$ storage. Moreover, if the solves with K can be preconditioned with an effective condition number independent of n and if the matrix-vector product can be carried out in $O(n)$ or $O(n \log n)$ time, then the computational effort is about $O(n)$ per optimization step. The desired preconditioning is achieved under limited circumstances by Stein, Chen and Anitescu (2012), whereas the fast matrix-vector product is attainable on regular grids by using circulant embedding or on irregular ones by fast multipole approximations of the Gaussian process kernel; see the work of Anitescu, Chen and Wang (2012). Second, if at any point one wants to obtain a better approximation to the score function, it suffices to consider additional U_j's in (1.2). However, how exactly to do this if using the dependent sampling scheme for the U_j's in Section 4 is not so obvious.

Since this approach provides only an approximation to the MLE, one must compare it with other possible approximations to the MLE. Many such approaches exist, including spectral methods, low-rank approximations, covariance tapering, and those based on some form of composite likelihood. All these methods involve computing the likelihood itself and not just its gradient, and thus all share this advantage over solving (1.2). Note that one can use randomized algorithms to approximate $\log\det K$ and thus approximate the loglikelihood directly (Zhang, 2006). However, this approximation requires first taking a power series expansion of K and then applying the randomization trick to each term in the truncated power series; the examples presented by Zhang (2006) show that the approach does not generally provide a good approximation to the loglikelihood. Since the accuracy of the power series approximation to $\log\det K$ depends on the condition number of K, some of the filtering ideas described by Stein, Chen and Anitescu (2012) and used to good effect in Section 4 here could perhaps be of value for approximating $\log\det K$, but we do not explore that possibility.

Let us consider the four approaches of spectral methods, low-rank approximations, covariance tapering, and composite likelihood in turn. Spectral approximations to the likelihood can be fast and accurate for gridded data (Whittle, 1954; Guyon, 1982; Dahlhaus and Künsch, 1987), although even for gridded data they may require some prefiltering to work well (Stein, 1995). In addition, the approximations tend to work less well as the number of dimensions increase (Dahlhaus and Künsch, 1987) and thus may be problematic for space-time data, especially if the number of spatial dimensions is three. Spectral approximations have been proposed for ungridded data (Fuentes, 2007), but they do not work as well as they do for gridded data from either a statistical or computational perspective, especially if large subsets of observations do not form a regular grid. Furthermore, in contrast to the approach we propose here, there appears to be no easy way of improving the approximations by doing further calculations, nor is it clear how to assess the loss of efficiency by using spectral approximations without a large extra computational burden.

Low-rank approximations, in which the covariance matrix is approximated by a low-rank matrix plus a diagonal matrix, can greatly reduce the burden of memory and computation relative to the exact likelihood (Cressie and Johannesson, 2008; Eidsvik et al., 2012). However, for the kinds of applications we have in mind, in which the diagonal component of the covariance matrix does not dominate the small-scale variation of the process, these low-rank approximations tend to work poorly and are not a viable option (Stein, 2007).

Covariance tapering replaces the covariance matrix of interest by a sparse covariance matrix with similar local behavior (Furrer, Genton and Nychka, 2006). There is theoretical support for this approach (Kaufman, Schervish and Nychka, 2008; Wang and Loh, 2011), but the tapered covariance matrix must be very sparse to help a great deal with calculating the log determinant of the covariance matrix, in which case, Stein (submitted) finds that composite likelihood approaches will often be preferable. There is scope for combining covariance tapering with the approach presented here in that sparse matrices lead to efficient matrix-vector multiplication, which is also essential for our implementation of computing (1.2) based on iterative methods to do the matrix solves. **?** show that covariance tapering and low-rank approximations can also sometimes be profitably combined to approximate likelihoods.

We consider methods based on composite likelihoods to be the main competitor to solving (1.2). The approximate loglikelihoods described by Vecchia (1988); Stein, Chi and Welty (2004); Caragea and Smith (2007) can all be written in the following form: for some sequence of pairs of matrices (A_j, B_j),

$j = 1, \ldots, q$ all with n columns, at most n rows and full rank,

$$\sum_{j=1}^{q} \log f_{j,\theta}(A_j Z \mid B_j Z), \tag{1.3}$$

where $f_{j,\theta}$ is the conditional Gaussian density of $A_j Z$ given $B_j Z$. As proposed by Vecchia (1988) and Stein, Chi and Welty (2004), the rank of B_j will generally be larger than that of A_j, in which case the main computation in obtaining (1.3) is finding Cholesky decompositions of the covariance matrices of $B_1 Z, \ldots, B_q Z$. For example, Vecchia (1988) just lets $A_j Z$ be the jth component of Z and $B_j Z$ some subset of $Z_1, \ldots, Z_{j-1}$. If m is the largest of these subsets, then the storage requirements for this computation are $O(m^2)$ rather than $O(n^2)$. Comparable to increasing the number of U_j's in the randomized algorithm used here, this approach can be updated to obtain a better approximation of the likelihood by increasing the size of the subset of $Z_1, \ldots, Z_{j-1}$ to condition on when computing the conditional density of Z_j. However, for this approach to be efficient from the perspective of flops, one needs to store the Cholesky decompositions of the covariance matrices of $B_1 Z, \ldots, B_q Z$, which would greatly increase the memory requirements of the algorithm. For dealing with truly massive datasets, our long-term plan is to combine the randomized approach studied here with a composite likelihood by using the randomized algorithms to compute the gradient of (1.3), thus making it possible to consider A_j's and B_j's of larger rank than would be feasible if one had to do exact calculations.

Section 2 provides a bound on the efficiency of the estimating equations based on the approximate likelihood relative to the Fisher information matrix. The bound is in terms of the condition number of the true covariance matrix of the observations and shows that if the covariance matrix is well-conditioned, N does not need to be very large to obtain nearly optimal estimating equations. Section 3 shows how one can get improved estimating equations by choosing the U_j's in (1.2) based on a design related to 2^n factorial designs. Section 4 describes details of the algorithms, including methods for solving the approximate score equations and the role of preconditioning. Section 5 provides results of numerical experiments on simulated data. These results show that the basic method can work well for moderate values of N, even sometimes when the condition numbers of the covariance matrices do not stay bounded as the number of observations increases. Furthermore, the algorithm with the U_j's chosen as in Section 3 can lead to substantially more accurate approximations for a given N. A large-scale numerical experiment shows that for observations on a partially occluded grid, the algorithm scales nearly linearly in the sample size. Section 6 applies the methods to the numerical solution of a set of partial differential equations describing fluid flow in a system with two dense bodies of different shapes sinking through the fluid. Because the fluid is treated as incompressible, the pressure field of the solution should be a harmonic function. Thus, the discrete Laplacian of the computed pressure field should be very nearly 0, at least away from the boundaries of the dense bodies. Within certain subdomains, a Gaussian process model appears to fit the filtered pressure field well, although the data show clear evidence that the spatial covariance structure differs within the two bodies, which may provide insight into the nature of the numerical errors of the solution to the PDE.

2. Variance of Stochastic Approximation of Score Function.

This section gives a bound relating the covariance matrices of the approximate and exact score functions. Write $g(\theta, N)$ for the random vector in $\mathbb{R}^p$ whose ith component is $g_i(\theta, N)$ as defined in (1.2). For a matrix M, denote its ijth element by M_{ij}. We can evaluate the effectiveness of the estimating equations in (1.2) by considering the $p \times p$ matrices $A(\theta)$ with $A_{ij} = E_\theta \frac{\partial}{\partial \theta_i} g_j(\theta, N)$ and $B(\theta)$ the covariance matrix of $g(\theta, N)$. General theory for estimating equations (Heyde, 1997) suggests that making the matrix $A(\theta)' B(\theta)^{-1} A(\theta)$ as large as possible in the ordering of positive semidefinite matrices is a natural criterion for assessing the statistical efficiency of estimating equations. One can easily show that $A_{ij}(\theta) = -\frac{1}{2}\mathrm{tr}(K^{-1}K_i K^{-1} K_j)$, so that $-A(\theta) = \mathcal{I}(\theta)$, the Fisher information matrix for θ based on Z (Stein, 1999, p. 179). Furthermore, writing W^i for $K^{-1}K_i$ and defining the matrix $\mathcal{J}(\theta)$ by $\mathcal{J}_{ij}(\theta) = \mathrm{cov}(U_1' W^i U_1, U_1' W^j U_1)$, we have

$$B_{ij}(\theta) = \mathcal{I}_{ij}(\theta) + \frac{1}{4N}\mathcal{J}_{ij}(\theta). \tag{2.1}$$

As $N \to \infty$, $A(\theta)' B(\theta)^{-1} A(\theta) \to \mathcal{I}(\theta)$, and this limit is what one gets for the exact score equations (1.1). Indeed, under sufficient regularity conditions on the model and the estimating equations, $\mathcal{I}(\theta) - A(\theta)' B(\theta)^{-1} A(\theta)$ is positive semidefinite (Bhapkar, 1972); hence, the score equations are, in this sense, generally the optimal unbiased estimating equations.

In fact, as also demonstrated empirically by Anitescu, Chen and Wang (2012), one may often not need N to be that large to get estimating equations that are nearly as efficient as the exact score equations. Writing U_{1j} for the

jth component of U_1, we have

$$\begin{aligned}
\mathcal{J}_{ij}(\theta) &= \sum_{k,\ell,p,q=1}^{n} \operatorname{cov}(W^i_{k\ell}U_{1k}U_{1\ell}, W^j_{pq}U_{1p}U_{1q}) \\
&= \sum_{k\neq\ell}\{\operatorname{cov}(W^i_{k\ell}U_{1k}U_{1\ell}, W^j_{k\ell}U_{1k}U_{1\ell}) + \operatorname{cov}(W^i_{k\ell}U_{1k}U_{1\ell}, W^j_{\ell k}U_{1k}U_{1\ell})\} \\
&= \sum_{k\neq\ell}(W^i_{k\ell}W^j_{k\ell} + W^i_{k\ell}W^j_{\ell k}) \\
&= \operatorname{tr}(W^iW^j) + \operatorname{tr}\{W^i(W^j)'\} - 2\sum_{k=1}^{n} W^i_{kk}W^j_{kk}. \qquad (2.2)
\end{aligned}$$

As noted by Hutchinson (1990), the terms with $k=\ell$ drop out in the second step because $U^2_{1j}=1$ with probability 1. When $K(\theta)$ is diagonal for all θ, then $N=1$ gives the exact score equations, although in this case computing $\operatorname{tr}(K^{-1}K_i)$ directly would be trivial.

For positive semidefinite matrices C and D write $C \preceq D$ if $D-C$ is positive semidefinite. We can bound $B(\theta)$ in terms of $\mathcal{I}(\theta)$ and the condition number of K.

Theorem 2.1.

$$B(\theta) \preceq \mathcal{I}(\theta)\left\{1 + \frac{(\kappa(K)+1)^2}{4N\kappa(K)}\right\}. \qquad (2.3)$$

The condition number enters the bound because the W^i's are not symmetric. If we instead write $\operatorname{tr}(K^{-1}K_i)$ as $\operatorname{tr}((G')^{-1}K_iG^{-1})$, where G is any square root of K as in $K=G'G$, we then have

$$h_i(\theta,N) = \frac{1}{2}Z'K^{-1}K_iK^{-1}Z - \frac{1}{2N}\sum_{j=1}^{N}U_j'(G')^{-1}K_iG^{-1}U_j = 0 \qquad (2.4)$$

for $i=1,\dots,p$ are also unbiased estimating equations for θ. In this case, the covariance matrix of the score function is just $\left(1+\frac{1}{N}\right)\mathcal{I}(\theta)$, which is less than or equal to the bound in (2.3) on $B(\theta)$. Whether it is preferable to use (2.4) rather than (1.2) depends on a number of factors including the sharpness of the bound in (1.2) and how much more work it takes to compute $G^{-1}U_j$ than to compute $K^{-1}U_j$. An example of how the action of such a square root can be approximated efficiently using only $O(n)$ storage is presented by Chen, Anitescu and Saad (2011).

3. Dependent Designs. Choosing the U_j's independently is simple and convenient, but one can reduce the variation in the stochastic approximation by using a more sophisticated design for the U_j's; this section describes such a design. Suppose that $n = Nm$ for some nonnegative integer m and that $\beta_1, \dots \beta_N$ are fixed vectors of length N with all entries ±1 for which $\frac{1}{N}\sum_{j=1}^{N}\beta_j\beta_j' = I$. For example, if $N = 2^q$ for a positive integer q, then the β_j's can be chosen to be the design matrix for a saturated model of a 2^q factorial design in which the levels of the factors are set at ±1 (Box, Hunter and Hunter, 2005, Ch. 5). In addition, assume that $X_1, \dots, X_m$ are random diagonal matrices of size N and Y_{jk}, $j = 1, \dots, N; k = 1, \dots, m$ are random variables such that all the diagonal elements of the X_j's and all the Y_{jk}'s are iid symmetric Bernoulli random variables. Then define

$$U_j = \begin{pmatrix} Y_{j1}X_1 \\ \vdots \\ Y_{jm}X_m \end{pmatrix} \beta_j. \tag{3.1}$$

One can easily show that for any $Nm \times Nm$ matrix M, $E\left(\frac{1}{N}\sum_{j=1}^{N} U_j' M U_j\right) = \mathrm{tr}(M)$. Thus, we can use this definition of the U_j's in (1.2), and the resulting estimating equations are still unbiased.

This design is closely related to a class of designs introduced by Avron and Toledo (2011), who propose selecting the U_j's as follows. Suppose H is a Hadamard matrix; that is, an $n \times n$ orthogonal matrix with elements ±1. Avron and Toledo (2011) actually consider H a multiple of a unitary matrix, but the special case H Hadamard makes their proposal most similar to ours. Then, using simple random sampling (with replacement), they choose N columns from this matrix and multiply this $n \times N$ matrix by an $n \times n$ diagonal matrix with diagonal entries made up of independent symmetric Bernoulli random variables. The columns of this resulting matrix are the U_j's. We are also multiplying a subset of the columns of a Hadamard matrix by a random diagonal matrix, but we do not select the columns by simple random sampling from some arbitrary Hadamard matrix.

The extra structure we impose yields beneficial results in terms of the variance of the randomized trace approximation as the following calculations show. Partitioning M into an $m \times m$ array of $N \times N$ matrices with $k\ell$th block $M^b_{k\ell}$, we obtain the following:

$$\frac{1}{N}\sum_{j=1}^{N} U_j' M U_j = \frac{1}{N}\sum_{k,\ell=1}^{m}\sum_{j=1}^{N} Y_{jk}Y_{j\ell}\beta_j' X_k M^b_{k\ell} X_\ell \beta_j. \tag{3.2}$$

Using $Y_{jk}^2 = 1$ and $X_k^2 = I$, we have

$$\begin{aligned}\frac{1}{N}\sum_{j=1}^{N} Y_{jk}^2 \beta_j' X_k M_{kk}^b X_k \beta_j &= \frac{1}{N}\mathrm{tr}\left(X_k M_{kk}^b X_k \sum_{j=1}^{N} \beta_j \beta_j' \right) \\ &= \mathrm{tr}(M_{kk}^b X_k^2) \\ &= \mathrm{tr}(M_{kk}^b),\end{aligned}$$

which is not random. Thus, if M is block diagonal (i.e., $M_{k\ell}^b$ is a matrix of zeroes for all $k \neq \ell$), (3.2) yields $\mathrm{tr}(M)$ without error. This result is an extension of the result that independent U_j's give $\mathrm{tr}(M)$ exactly for diagonal M. Furthermore, it turns out that, at least in terms of the variance of $\frac{1}{N}\sum_{j=1}^{N} U_j' M U_j$, for the elements of M off the block diagonal, we do exactly the same as we do when the U_j's are independent. Define $B^d(\theta)$ to be the covariance matrix of $g(\theta, N)$ as defined by (1.2) but with the U_j's defined by (3.1) and take $T(N, n)$ to be the set of pairs of positive integers (k, ℓ) with $1 \leq \ell < k \leq n$ for which $\lfloor k/N \rfloor = \lfloor \ell/N \rfloor$. We have the following inequality, whose proof is given in the Appendix.

THEOREM 3.1. *For any vector* $v = (v_1, \dots, v_p)'$,

$$v' B(\theta) v - v' B^d(\theta) v = \frac{2}{N} \sum_{(k,\ell)\in T(N,n)} \left\{ \sum_{i=1}^{N} v_i \big(W_{k\ell}^i + W_{\ell k}^i\big) \right\}^2. \tag{3.3}$$

Thus, $B(\theta) \succeq B^d(\theta)$, and the U_j's defined by (3.1) always yield a more efficient set of estimating equations than do independent U_j's.

How much of an improvement will result from using dependent U_j's depends on the size of the $W_{k\ell}^i$'s within each block. For spatial data, one would typically group spatially contiguous observations within blocks. How to block for space-time data is less clear. The results here focus on the variance of the randomized trace approximation. Avron and Toledo (2011) obtain bounds on the probability that the approximation error is less than some quantity and note that these results sometimes give rankings for various randomized trace approximations different from those obtained by comparing variances.

4. Computational Aspects. Numerically solving the estimating equations (1.2) requires an outer nonlinear equation solver and an inner linear equation solver. The nonlinear solver starts at an initial guess θ^0 and iteratively updates it to approach the zero of (1.2). In each iteration, at θ^i, the nonlinear solver typically requires an evaluation of $g(\theta^i, N)$ in order to find the next iterate θ^{i+1}. In turn, the evaluation of g requires employing a linear solver to compute the set of vectors $K^{-1}Z$ and $K^{-1}U_j$, $j = 1, \dots, N$.

The Fisher information matrix $\mathcal{I}(\theta)$ and the matrix $\mathcal{J}(\theta)$ contain terms involving matrix traces and diagonals. Write $\operatorname{diag}(\cdot)$ for a column vector containing the diagonal elements of a matrix, and $\circ$ for the Hadamard (elementwise) product of matrices. For any real matrix A,

$$\operatorname{tr}(A) = E_U(U'AU) \quad \text{and} \quad \operatorname{diag}(A) = E_U(U \circ AU),$$

where the expectation E_U is taken over U, a random vector with iid symmetric Bernoulli components. One can unbiasedly estimate $\mathcal{I}(\theta)$ and $\mathcal{J}(\theta)$ by

$$\mathcal{I}_{ij}(\theta) \approx \frac{1}{2N_2} \sum_{k=1}^{N_2} U_k' W^i W^j U_k$$

and

$$\begin{aligned} \mathcal{J}_{ij}(\theta) \quad &\approx \quad \frac{1}{N_2} \sum_{k=1}^{N_2} U_k' W^i W^j U_k + \frac{1}{N_2} \sum_{k=1}^{N_2} U_k' W^i (W^j)' U_k \\ &\quad - \frac{2}{N_2} \sum_{k=1}^{N_2} (U_k \circ W^i U_k)' (U_k \circ W^j U_k). \end{aligned}$$

Note that here the set of vectors U_k need not be the same as that in (1.2), but we use the same notation for simplicity. In this approximation, evaluating $\mathcal{I}(\theta)$ and $\mathcal{J}(\theta)$ also requires linear solves since $W^i U_k = K^{-1}(K_i U_k)$ and $(W^i)' U_k = K_i(K^{-1}U_k)$.

4.1. *Linear Solver.* We consider an iterative solver for solving a set of linear equations $Ax = b$ for a symmetric positive definite matrix $A \in \mathbb{R}^{n \times n}$, given a right-hand vector b. Since the matrix A (in our case the covariance matrix) is symmetric positive definite, the conjugate gradient algorithm is naturally used. Let x^i be the current approximate solution, and let $r^i = b - Ax^i$ be the residual. The algorithm finds a search direction q^i and a step size α^i to update the approximate solution, that is, $x^{i+1} = x^i + \alpha^i q^i$, such that the search directions $q^i, \dots, q^0$ are mutually A-conjugate (i.e., $(q^i)' A q^j = 0$ for $i \neq j$) and the new residual r^{i+1} is orthogonal to all the previous ones, $r^i, \dots, r^0$. One can show that the search direction is a linear combination of the current residual and the past search direction, yielding the following recurrence formulas:

$$\begin{aligned} x^{i+1} &= x^i + \alpha^i q^i, \\ r^{i+1} &= r^i - \alpha^i A q^i, \\ q^{i+1} &= r^{i+1} + \beta^i q^i, \end{aligned}$$

where $\alpha^i = \left\langle r^i, r^i \right\rangle / \left\langle A q^i, q^i \right\rangle$ and $\beta^i = \left\langle r^{i+1}, r^{i+1} \right\rangle / \left\langle r^i, r^i \right\rangle$, and $\langle \cdot, \cdot \rangle$ denotes the vector inner product. Letting x^* be the exact solution, that is, $Ax^* = b$, then x^i enjoys a linear convergence to x^*:

$$\|x^i - x^*\|_A \le 2 \left(\frac{\sqrt{\kappa(A)} - 1}{\sqrt{\kappa(A)} + 1} \right)^i \|x^0 - x^*\|_A, \tag{4.1}$$

where $\| \cdot \|_A = \langle A \cdot, \cdot \rangle^{\frac{1}{2}}$ is the A-norm of a vector.

Asymptotically, the time cost of one iteration is upper bounded by that of multiplying A by q^i, which typically dominates other vector operations when A is not sparse. Properties of the covariance matrix can be exploited to efficiently compute the matrix-vector products. For example, when the observations are on a lattice (regular grid), one can use the fast Fourier transform (FFT), which takes time $O(n \log n)$ (Chan and Jin, 2007). Even when the grid is partial (with occluded observations), this idea can still be applied. On the other hand, for nongridded observations, one can use a combination of direct summations for close-by points and multipole expansions of the covariance kernel for faraway points to compute the matrix-vector products in $O(n \log n)$, even $O(n)$, time (Barnes and Hut, 1986; Greengard and Rokhlin, 1987). In the case of Matérn-type Gaussian processes and in the context of solving the stochastic approximation (1.2), such fast multipole approximations were presented by Anitescu, Chen and Wang (2012). Note that the total computational cost of the solver is the cost of each iteration times the number of iterations, the latter being usually much less than n.

The number of iterations to achieve a desired accuracy depends on how fast x^i approaches x^*, which, from (4.1), is in turn affected by the condition number κ of A. Two techniques can be used to improve convergence. One is to perform preconditioning in order to reduce κ; this technique will be discussed in the next section. The other is to adopt a block version of the conjugate gradient algorithm. This technique is useful for solving the linear system for the same matrix with multiple right-hand sides. Specifically, denote by $AX = B$ the linear system one wants to solve, where B is a matrix with s columns, and the same for the unknown X. Conventionally, matrices such as B are called *block vectors*, honoring the fact that the columns of B are handled simultaneously. The block conjugate gradient algorithm is similar to the single-vector version except that the iterates x^i, r^i and q^i now become block iterates X^i, R^i, and Q^i and the coefficients α^i and β^i become $s \times s$ matrices. The detailed algorithm is not shown here; interested readers are referred to O'Leary (1980). If X^* is the exact solution, then X^i approaches X^* linearly:

$$\|(X^i)_j - (X^*)_j\|_A \leq C_j \left(\frac{\sqrt{\kappa_s(A)} - 1}{\sqrt{\kappa_s(A)} + 1} \right)^i, \qquad j = 1, \ldots, s, \tag{4.2}$$

where $(X^i)_j$ and $(X^*)_j$ are the jth column of X^i and X^*, respectively; C_j is some constant dependent on j but not i; and $\kappa_s(A)$ is the ratio between $\lambda_n(A)$ and $\lambda_s(A)$ with the eigenvalues λ_k sorted increasingly. Comparing (4.1) with (4.2), we see that the modified condition number κ_s is less than κ, which means that the block version of the conjugate gradient algorithm has a faster convergence than the standard version does. In practice, since there are many right-hand sides (i.e., the vectors Z, U_j's and K_iU_k's), we always use the block version.

4.2. *Preconditioning/Filtering.* Preconditioning is a technique for reducing the condition number of the matrix. Here, the benefit of preconditioning is twofold: it encourages the rapid convergence of an iterative linear solver, and, if the effective condition number is small, it strongly bounds the uncertainty in using the estimating equations (1.2) instead of (1.1) for estimating parameters (see Theorem 2.1). In numerical linear algebra, preconditioning refers to applying a matrix M, which approximates the inverse of A in some sense, to both sides of the linear system of equations. In the simple case of left preconditioning, this amounts to solving $MAx = Mb$ for MA better-conditioned than A. With certain algebraic manipulations, the matrix M enters into the conjugate gradient algorithm in the form of multiplication with vectors. For the detailed algorithm, see Saad (2003). This technique does not explicitly compute the matrix MA, but it requires that the matrix-vector multiplications with M can be efficiently carried out.

For covariance matrices, certain filtering operations are known to reduce the condition number, and some can even achieve an optimal preconditioning in the sense that the condition number is bounded by a constant independent of the size of the matrix (Stein, Chen and Anitescu, 2012). Note that these filtering operations may or may not preserve the rank/size of the matrix. When the rank is reduced, then some loss of statistical information results when filtering, although similar filtering is also likely needed to apply spectral methods for strongly correlated spatial data on a grid (Stein, 1995).

Therefore, we consider applying the same filter to all the vectors and matrices in the estimating equations, in which case, (1.2) becomes the stochastic approximation to the score equations of the *filtered* process. Evaluating the filtered version of $g(\theta, N)$ becomes easier because the linear solves with the filtered covariance matrix converge faster.

4.3. *Nonlinear Solver.* The choice of the outer nonlinear solver is problem dependent. The purpose of solving the score equations (1.1) or the estimating equations (1.2) is to maximize the loglikelihood function $\mathcal{L}(\theta)$. Therefore, investigation into the shape of the loglikelihood surface helps identify an appropriate solver.

In this paper, we consider the power law generalized covariance model ($\alpha > 0$):

$$G(x;\theta) = \begin{cases} \Gamma(-\alpha/2)r^\alpha, & \text{if } \alpha/2 \notin \mathbb{N} \\ (-1)^{1+\alpha/2} r^\alpha \log r, & \text{if } \alpha/2 \in \mathbb{N} \end{cases}$$

where $x = [x_1, \dots, x_d] \in \mathbb{R}^d$ denotes coordinates, θ is the set of parameters containing $\alpha > 0$, $\ell = [\ell_1, \dots, \ell_d] \in \mathbb{R}^d$, and r is the elliptical radius

$$r = \sqrt{\frac{x_1^2}{\ell_1^2} + \cdots + \frac{x_d^2}{\ell_d^2}}.$$

Allowing a different scaling in different directions may be appropriate when, for example, variations in a vertical direction may be different from those in a horizontal direction. The function G is conditionally positive definite; therefore, only the covariances of authorized linear combinations of the process are defined (Chilés and Delfiner, 1999, Sec. 4.3). In fact, G is p-conditionally positive definite if and only if $2p + 2 > \alpha$ (see Chilés and Delfiner, 1999, Sec. 4.4), so that applying the discrete Laplace filter (which gives second-order differences) at least $\lceil \alpha/2 \rceil$ times to the observations yields a set of authorized linear combinations. Stein, Chen and Anitescu (2012) show that if a discrete Laplace filter is applied τ times and $\alpha + d = 4\tau$, the covariance matrix has a bounded condition number independent of the matrix size. Therefore, if the grid is $\{\delta \boldsymbol{j}\}$ for some fixed spacing δ and $\boldsymbol{j}$ a vector whose components take integer values between 0 and m, then applying the filter $\tau = \text{round}((\alpha + d)/4)$ times, we obtain the covariance matrix

$$K_{\boldsymbol{ij}} = \text{cov}\{\Delta^\tau Z(\delta \boldsymbol{i}), \Delta^\tau Z(\delta \boldsymbol{j})\},$$

where Δ denotes the discrete Laplace operator

$$\Delta Z(\delta \boldsymbol{j}) = \sum_{p=1}^{d} \{Z(\delta \boldsymbol{j} - \delta \boldsymbol{e}_p) - 2Z(\delta \boldsymbol{j}) + Z(\delta \boldsymbol{j} + \delta \boldsymbol{e}_p)\},$$

with $\boldsymbol{e}_p$ meaning the unit vector along the pth coordinate. The resulting K is both positive definite and well-conditioned.

Figure 1 shows a sample loglikelihood surface for $d = 1$ based on an observation vector Z simulated from a 1D partial regular grid spanning the range $[0, 100]$, using parameters $\alpha = 1.5$ and $\ell = 10$. (A similar 2D grid is shown later in Figure 2.) The peak of the surface is denoted by the solid white dot, which is not far away from the truth $\theta = (1.5, 10)$. The white dashed curve (profile of the surface) indicates the maximum loglikelihoods $\mathcal{L}$ given α. The curve is also projected on the $\alpha - \mathcal{L}$ plane and the $\alpha - \ell$ plane. One sees that the loglikelihood value has small variation (ranges from 48 to 58) along this curve compared with the rest of the surface, whereas the parameter ℓ changes the likelihood substantially.

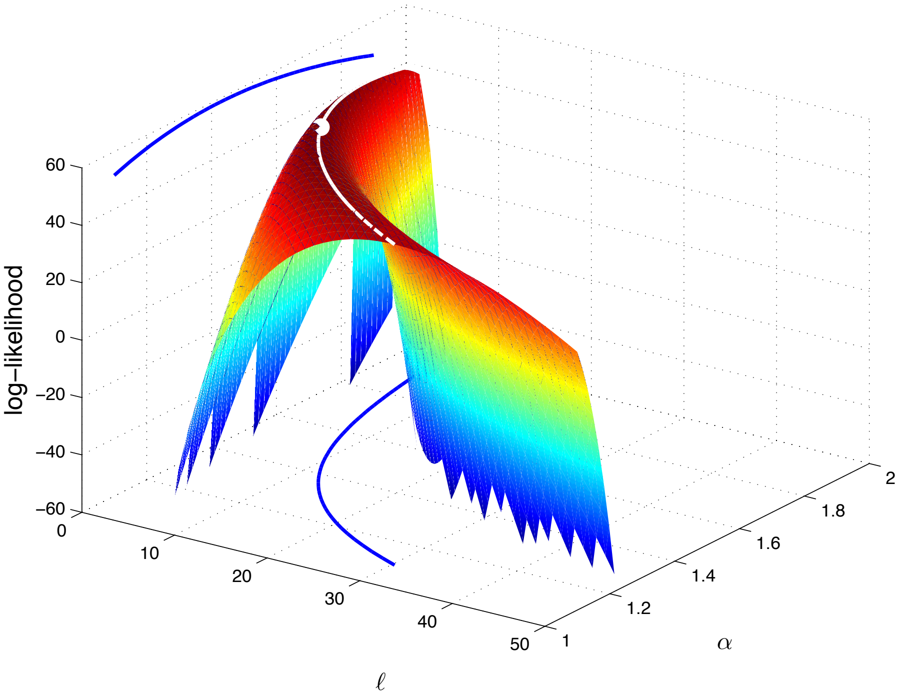


Fig 1. *A sample loglikelihood surface for the power law generalized covariance kernel, with profile curve and peak plotted.*

A Newton-type nonlinear solver starts at some initial point θ^0 and tries to approach the optimal point (one that solves the score equations).[1] Let the

[1] To facilitate understanding, we explain here the process for solving the score equations (1.1). Conceptually it is similar to that for solving the estimating equations (1.2).

current point be θ^i; then the solver finds a direction q^i (typically the inverse of the Jacobian multiplied by θ^i, that is, $q^i = \nabla_\theta g(\theta^i, N)^{-1}\theta^i$) and a step size α^i in some way to increase the value of $\mathcal{L}$ evaluated at $\theta^{i+1} = \theta^i + \alpha^i q^i$ or, equivalently, is closer to a solution of the score equations. Corresponding to Figure 1, the solver starts somewhere on the surface and quickly climbs to a point along the profile curve. However, this point might be far away from the peak. It turns out that along this curve a Newton-type solver is usually unable to find a direction with an appropriate step size to numerically increase $\mathcal{L}$, in part because of the narrow ridge indicated in the figure. The variation of $\mathcal{L}$ along the normal direction of the curve is much larger than that along the tangent direction. Thus, the iterate θ^i is trapped and cannot advance to the peak. In such a case, even though the estimate of α and the score function is reasonably close to its solution, the estimate of ℓ could be erroneous.

To successfully solve the estimating equations, we consider each component of ℓ an implicit function of α. Denote by

$$g_i(\ell_1, \dots, \ell_d, \alpha) = 0, \qquad i = 1, \dots, d+1, \tag{4.3}$$

the estimating equations, ignoring the fixed variable N. The implicit function theorem indicates that a set of functions $\ell_1(\alpha), \dots, \ell_d(\alpha)$ exists around an isolated zero of (4.3) in a neighborhood where (4.3) is continuously differentiable, such that

$$g_i(\ell_1(\alpha), \dots, \ell_d(\alpha), \alpha) = 0, \quad \text{for} \quad i = 2, \dots, d+1.$$

Therefore, we need only to solve the equation

$$g_1(\ell_1(\alpha), \dots, \ell_d(\alpha), \alpha) = 0 \tag{4.4}$$

with a single variable α. Numerically, a much more robust method than a Newton-type method exists for finding a root of a one-variable function. We use the method of Forsythe, Malcolm and Moler (1976) for solving (4.4). This method in turn requires the evaluation of the left-hand side of (4.4). Then, the ℓ_i's are evaluated by solving $g_2, \dots, g_{d+1} = 0$ fixing α, whereby a Newton-type algorithm is empirically proven to be an efficient method.

5. Experiments.

In this section, we show a few experimental results based on a partially occluded regular grid. The rationale for using such a partial grid is to illustrate a setting where spectral techniques do not work so well but efficient matrix-vector multiplications are available. A partially occluded grid can occur, for example, when observations of some surface characteristics are taken by a satellite-based instrument and it is not possible to obtain observations over regions with sufficiently dense cloud cover. The grid has a physical range $[0, 100] \times [0, 100]$, with a hole in a disc shape of radius 10 located at $(40, 60)$. An illustration of the grid, with size 32×32, is shown in Figure 2. The matrix-vector multiplication is performed by first doing the multiplication using the full grid via circulant embedding and FFT, followed by removing the entries corresponding to the hole of the grid. Recall that the covariance model is defined in Section 4.3, along with the explanation of the filtering step.

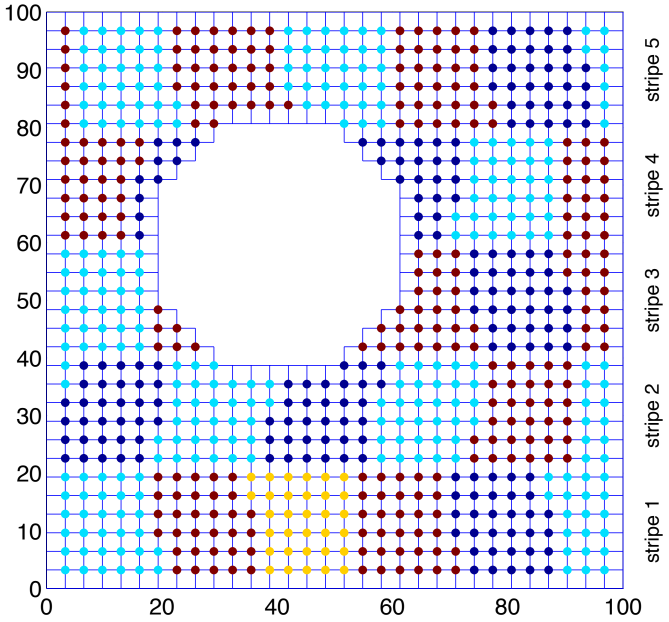


Fig 2. *A 32×32 grid with a region of missing observations in a disc shape.*

When working with dependent samples, it is advantageous to group nearby grid points such that the resulting blocks have a plump shape and that there are as many blocks with size exactly N as possible. For an occluded grid, this is a nontrivial task. Here we use a simple heuristic to effectively group the points. We divide the grid into horizontal stripes of width $\lfloor\sqrt{N}\rfloor$ (in case $\lfloor\sqrt{N}\rfloor$ does not divide the grid size along the vertical direction, some stripes have a width $\lfloor\sqrt{N}\rfloor + 1$). The stripes are ordered from bottom to top, and the grid points inside the odd-numbered stripes are ordered lexicographically in their coordinates, that is, (x, y). In order to obtain as many contiguous blocks as possible, the grid points inside the even-numbered stripes are ordered lexicographically according to $(-x, y)$. This ordering gives a zigzag flow of the points starting from the bottom-left corner of the grid. Every N points are grouped in a block. The coloring of the grid points in Figure 2 shows an example of the grouping. Note that because of filtering, observations on either an external or internal boundary are not part of any block.

5.1. *Choice of N.* One of the most important factors that affect the efficacy of approximating the score equations is the value N. Theorem 2.1 indicates that N should increase at least like $\kappa(K)$ in order that the additional uncertainty introduced by approximating the score equations be comparable with that caused by the randomness of the sample Z. In the ideal case, when the condition number of the matrix (possibly with filtering) is bounded independent of the matrix size n, then even taking $N = 1$ is sufficient to obtain estimates with the same rate of convergence as the exact score equations. When κ grows with n, however, a better guideline for selecting N is to consider the growth of $\mathcal{I}^{-1}\mathcal{J}$.

Figure 3 plots the condition number and the norm of $\mathcal{I}^{-1}\mathcal{J}$ for varying sizes of the matrix. Although performing a Laplacian filtering will yield provably bounded condition numbers only for the case $\alpha = 2$, one sees that the filtering is also effective for the cases $\alpha = 1$ and 1.5. Moreover, the norm of $\mathcal{I}^{-1}\mathcal{J}$ is significantly smaller than κ when n is large, and in fact it does not seem to grow with n. This result indicates the bound in Theorem 1 is sometimes far too conservative and that using a fixed N can be effective even when κ grows with n.

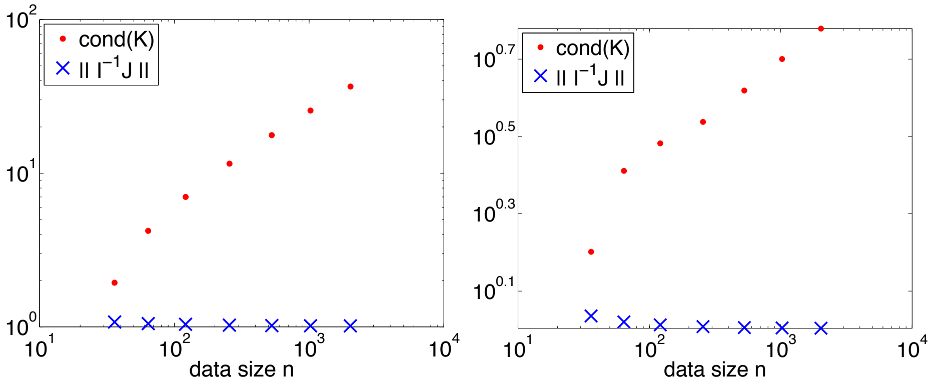


FIG 3. *Growth of κ compared with that of $\|\mathcal{I}^{-1}\mathcal{J}\|$, for power law kernel in 2D. Left: $\alpha = 1$; right: $\alpha = 1.5$.*

Of course, the norm of $\mathcal{I}^{-1}\mathcal{J}$ is not always bounded. In Figure 4 we show two examples using the Matérn covariance kernel with smoothness parameter $\nu = 1$ and 1.5 (essentially $\alpha = 2$ and 3). Without filtering, both $\kappa(K)$ and $\|\mathcal{I}^{-1}\mathcal{J}\|$ grow with n, although the plots show that the growth of the latter is significantly slower than that of the former.

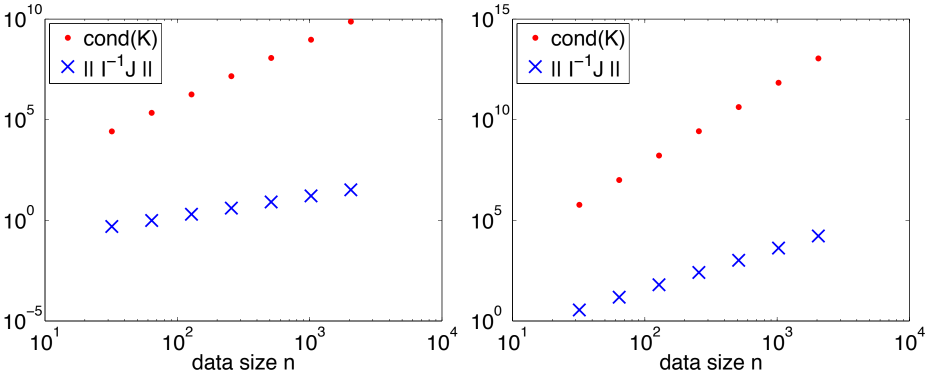
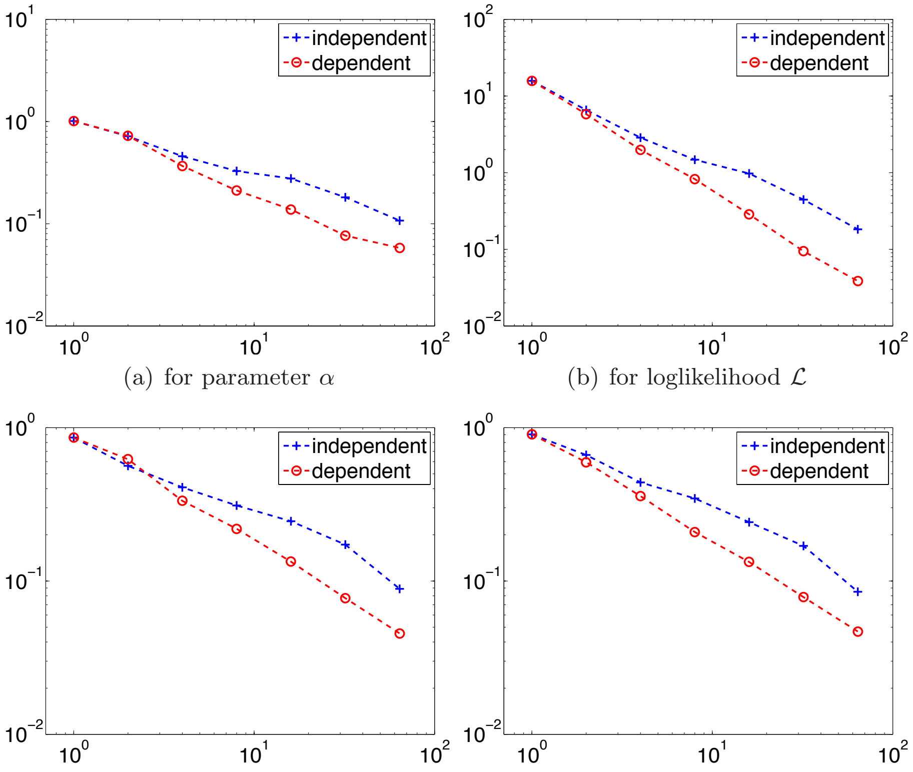


FIG 4. *Growth of κ compared with that of $\|\mathcal{I}^{-1}\mathcal{J}\|$, for Matérn kernel in 1D, without filtering. Left: $\nu = 1$; right: $\nu = 1.5$.*

5.2. *A 32×32 Grid Example.* Here, we show the details of solving the estimating equations (1.2) using a 32×32 grid as an example. Setting the truth $\alpha = 1.5$ and $\ell = (7, 10)$ (that is, $\theta = (1.5,\ 7,\ 10)$), consider exact and approximate maximum likelihood estimation based on the data obtained by applying the Laplacian filter once to the observations. One way to evaluate the approximate MLEs is to compute the ratios of the square roots of the diagonal elements of V^{-1}, where $V = A(\theta)'B(\theta)^{-1}A(\theta)$, to the square roots of the diagonal elements of $\mathcal{I}^{-1}$. We know these ratios must be at least 1, and that the closer they are to 1, the more nearly optimal the resulting estimating equations based on the approximate score function are. For $N = 64$ and independent sampling, we get 1.0156, 1.0125, and 1.0135 for the three ratios, all of which are very close to 1. Since one generally cannot calculate V^{-1} exactly, it is also worthwhile to compare a stochastic approximation of the diagonal values of V^{-1} to their exact values. When this was done once for $N = 64$ and by using $N_2 = 100$ in the approximation, the three ratios obtained were 0.9821, 0.9817, and 0.9833, which are all close to 1.

Figure 5 shows the performance of the resulting estimates. For $N = 1$, 2, 4, 8, 16, 32, and 64, we simulated 100 realizations of the process on the 32×32 occluded grid, applied the discrete Laplacian once, and then computed exact MLEs and approximations using both independent and dependent (as described in the beginning of Section 5) sampling. When $N = 1$, the independent and dependent sampling schemes are identical, so only results for independent sampling are given. Figure 5 plots, for each component of θ, the mean squared differences between the approximate and exact MLEs divided by the mean squared errors for the exact MLE's. As expected, these ratios decrease with N, particularly for dependent sampling. Indeed, dependent sampling is much more efficient than is independent sampling for larger N; for example, the results in Figure 5 show that dependent sampling with $N = 32$ yields better estimates for all three parameters than does independent sampling with $N = 64$.



(a) for parameter α

(b) for loglikelihood $\mathcal{L}$

(c) for parameter ℓ_1

(d) for parameter ℓ_2

FIG 5. *Effects of N (1, 2, 4, 8, 16, 32, 64). In each plot, the curve with the plus sign corresponds to the independent design, whereas that with the circle sign corresponds to the dependent design. The horizontal axis represents N. In plots (a), (c), and (d), the vertical axis represents the mean squared differences between the approximate and exact MLEs divided by the mean squared errors for the exact MLEs, for the components α, ℓ_1, and ℓ_2, respectively. In plot (b), the vertical axis represents the mean squared difference between the approximate and exact loglikelihood value.*

5.3. *Large-Scale Experiments.* We experimented with larger grids (in the same physical range). We show the results in Table 1 and Figure 6 for $N = 64$. When the matrix becomes large, we are unable to compute $\mathcal{I}$ and V exactly. Based on the preceding experiment, it seems reasonable to use $N_2 = 100$ in approximating $\mathcal{I}$ and V. Therefore, the eigenvalues in Table 1 were computed only approximately.

TABLE 1
Estimates and estimated standard errors for increasingly dense grids. The last three rows show the ratio of standard errors of the approximate to the exact MLE's.

Grid size	32×32	64×64	128×128	256×256	512×512	1024×1024
$\widehat{\theta}^N$	1.5355	1.5084	1.4919	1.4975	1.5011	1.5012
	6.8507	6.9974	7.1221	7.0663	6.9841	6.9677
	9.2923	10.062	10.091	10.063	9.9818	9.9600
$\sqrt{\lambda(\mathcal{I}^{-1})}$	0.0393	0.0125	0.0045	0.0018	0.0007	0.0003
	0.3231	0.1515	0.0732	0.0360	0.0179	0.0089
	0.9588	0.6599	0.4257	0.2621	0.1566	0.0912
$\dfrac{\sqrt{\lambda(V^{-1})}}{\sqrt{\lambda(\mathcal{I}^{-1})}}$	1.0016	1.0008	1.0004	1.0003	1.0002	1.0001
	1.0076	1.0077	1.0077	1.0077	1.0077	1.0077
	1.0063	1.0070	1.0073	1.0074	1.0075	1.0076

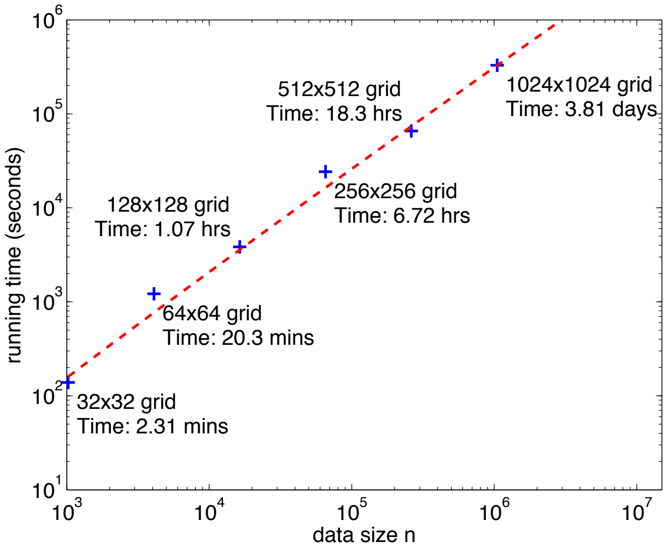


FIG 6. *Running time for increasingly dense grids.*

One sees that as the grid becomes larger (denser), the variance of the estimates decreases as expected. The matrices $\mathcal{I}^{-1}$ and V^{-1} are comparable in all cases, and in fact the ratios stay roughly the same across different sizes of the data. The experiments were run for data size up to around one million, and the scaling of the running time versus data size is favorable. The dashed curve in Figure 6 is a fit to the recorded times using a function in the form of $n \log n$ times a constant. One sees a strong agreement of the recorded times with the scaling $O(n \log n)$.

6. Application.

Fluid flows, such as ice melt in water or molten rock rising from the Earth's lower mantle, are described by partial differential equations (Furuichi, May and Tackley, 2011). Simulations can be conducted to study the dynamics and to enrich our understanding of the behavior of fluids. Nevertheless, finite-dimensional approximations of such phenomena mean that their computation inevitably contains errors, which in the case of large, multidimensional problems may be quite large when combined with limited computational resources. The effects of such numerical errors on the output must be quantified to achieve a predictive simulation capability. The typical numerical analysis thought process attempts to bound such errors using asymptotic considerations that in many circumstances results in excessively conservative bounds. In order to circumvent such shortcomings, new statistical-based approaches have been developed for quantifying numerical and other approximation errors (Glimm et al., 2003; Lee, 2005). In this vein, we investigate stochastic models for the residual error of phenomena modeled by partial differential equations that are subsequently approximated in a finite-dimensional space by finite-element techniques.

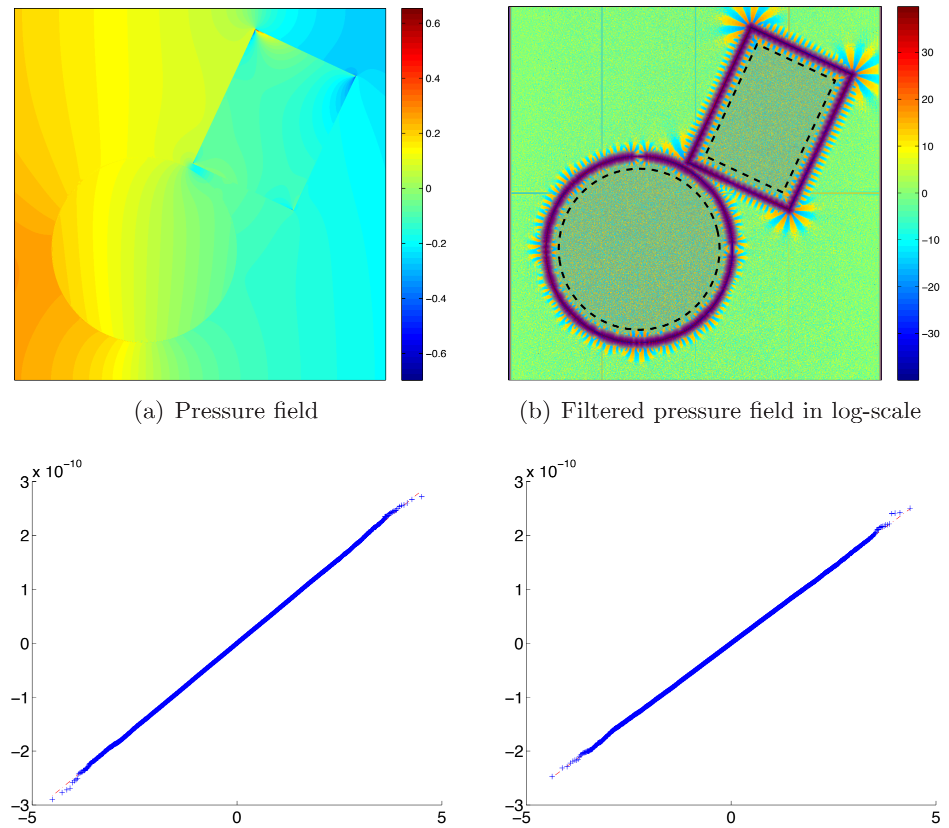
To this end, we study a set of geodynamics data simulated from the variable coefficient Stokes flow:

$$-\nabla \cdot (\eta D\boldsymbol{u} - p\mathbf{1}) - \rho \boldsymbol{g} = 0,$$
$$\nabla \cdot \boldsymbol{u} = 0,$$

where $\boldsymbol{u}$, the velocity and p, the pressure are unknowns, η is the given effective viscosity, $\rho\boldsymbol{g}$ is the gravitational body force, and D denotes the symmetric gradient, that is, $D\boldsymbol{u} = \frac{1}{2}(\nabla\boldsymbol{u} + (\nabla\boldsymbol{u})')$. Figure 7(a) shows the computed pressure on the square $[0, 100] \times [0, 100]$ discretized by a 1024×1024 grid.[2] This simulates the situation where two dense bodies, a disk and a rotated rectangle, with high viscosity ($\rho = 1$, $\eta = 100$) are sinking in a background fluid with low density and viscosity ($\rho = 0.2$, $\eta = 1$). This

[2] The system is solved by using a finite-element method with Q_1-Q_1 elements, stabilized by the technique of Dohrmann and Bochev (2004). The conservation properties of this discretization may not be sufficient for serious geodynamics simulations, but it is simple and serves our purposes. Homogeneous Dirichlet boundary conditions are imposed at all walls.

problem is a version of the SINKER benchmark problem (May and Moresi, 2008).



(a) Pressure field

(b) Filtered pressure field in log-scale

(c) Data in circle (prescribed by the black, dashed circular boundary)

(d) Data in rectangle (prescribed by the black, dashed rectangular boundary)

FIG 7. *Stokes flow data. The bottom plots are QQ plots of the filtered data inside the two dense bodies.*

Because the fluid is treated as incompressible, an exact solution of the equations is harmonic, so its Laplacian is 0. We should therefore expect the discrete Laplacian of the numerical results to show little variation, and this is indeed the case. The majority of the filtered data is close to zero, whereas large variations occur on the boundary of the two bodies, at which pressure can change rapidly.

The filtered data range over several orders of magnitude, so Figure 7(b) plots the rescaled logarithms of the absolute values of the filtered results. One clearly sees the large variations on the boundaries of the two bodies, and hence a stationary Gaussian process model for the filtered output over the entire region makes no sense. However, within the two dense bodies, as long as one stays sufficiently far from their boundaries, a stationary Gaussian process model for the filtered results may be appropriate. Figures 7(c) and (d) show the QQ-plots of the filtered data inside these two regions. The plots indicate remarkable agreement with a normal distribution, which supports the modeling of the filtered data using a Gaussian process. Plots of bivariate distributions (not shown) also show good agreement with a stationary Gaussian process model.

The extremely small values for the filtered data in Figures 7(c) and (d) show that the discrete Laplacian of the observations within the specified regions are nearly 0. Thus, it would not make sense to model the unfiltered observations using a model that did not take account of the near harmonic behavior of the data. Therefore, in this setting, filtering the data with the discrete Laplacian is not just a statistical or computational convenience to yield better conditioned covariance matrices; it is essential to obtaining a reasonable model for the data.

We fitted two models. In the first model, the (unfiltered) data has a mean function whose discrete Laplacian is 0 and whose generalized covariance function is the power-law function as described in Section 4.3:

$$G_1(x;\alpha,\theta_0) = \theta_0 \cdot \Gamma(-\alpha/2)|x|^\alpha.$$

The second model adds a nugget effect to the generalized covariance function:

$$G_2(x;\alpha,\theta_0,\theta_1) = \theta_0 \cdot [\ \Gamma(-\alpha/2)|x|^\alpha + \theta_1 \mathbf{1}_{x=0}\ ].$$

Table 2 shows the fitted results using $N = 16$ sampling vectors in the estimating equations. Since the filtered data Z is small, we pre-scaled Z by 6×10^{12} before the fitting. The fitted parameter θ_0 corresponds to the scaled data. The ratios between $\sqrt{\lambda(V^{-1})}$ and $\sqrt{\lambda(\mathcal{I}^{-1})}$ indicate that using $N = 16$ is sufficient to approximate the score equations.

The square roots of the eigenvalues of the inverse of the Fisher matrix provide approximate standard errors for the estimates. In all cases, these standard errors are small compared with the estimates themselves, indicating that all parameters are estimated well. Not surprisingly, the relative standard errors for $\widehat{\alpha}$ and $\widehat{\theta}_0$ are much larger when a nugget effect is included in the model.

We also compared the smoothness parameter α across different models and different regions. In the power-law model G_1, the values of α are small in both regions, suggesting that the underlying processes are extremely rough.

Table 2

Estimates and estimated standard errors for Stokes flow data.

Data	Circle	Rectangle	Circ.+Rect. (independent)
# points	1.5×10^5	7.9×10^4	2.3×10^5
Model G_1:			
$\widehat{\theta}^N = \begin{pmatrix} \alpha \\ \theta_0 \end{pmatrix}$	0.1819	0.3015	
	681.92	1177.2	
$\sqrt{\lambda(\mathcal{I}^{-1})}$	5.0421×10^{-4}	9.8030×10^{-4}	
	$1.3101 \times 10^{+1}$	$1.8553 \times 10^{+1}$	
$\frac{\sqrt{\lambda(V^{-1})}}{\sqrt{\lambda(\mathcal{I}^{-1})}}$	1.0011	1.0013	
	1.0278	1.0276	
Model G_2:			
$\widehat{\theta}^N = \begin{pmatrix} \alpha \\ \theta_0 \\ \theta_1 \end{pmatrix}$	2.1316	3.4353	2.6059
	5336.9	8677.6	7130.1
	0.9092	0.4371	0.6325
$\sqrt{\lambda(\mathcal{I}^{-1})}$	2.0741×10^{-2}	4.2246×10^{-2}	1.9346×10^{-2}
	$1.5558 \times 10^{+2}$	$1.9206 \times 10^{+2}$	$1.4942 \times 10^{+2}$
	3.5081×10^{-3}	2.2745×10^{-3}	1.9564×10^{-3}
$\frac{\sqrt{\lambda(V^{-1})}}{\sqrt{\lambda(\mathcal{I}^{-1})}}$	1.0283	1.0289	1.0284
	1.0284	1.0284	1.0284
	1.0010	1.0009	1.0010

Such small fitted values of α are not uncommon when a nugget effect is omitted from a model when it is needed. When the nugget effect is included, much larger estimated values of α are obtained. Specifically, $\widehat{\alpha}$ is then between 2 and 4 in both regions, which corresponds to a model for the observations of a function whose discrete Laplacian is 0 plus a random process with exactly one derivative in any direction plus a white noise term.

The last column of Table 2 fits G_2 to both regions simultaneously assuming independence between the processes in the two regions. This model is a submodel of the model in which the parameter values are allowed to be different in the two regions (and, again, independence across regions is assumed), so we could consider using a likelihood ratio test to assess whether the model with common parameters provides an adequate description of the data relative to the model with separate parameters. We now run into a problem with only having derivatives of the loglikelihood and not the loglikelihood itself. We could in principle recover the difference in loglikelihoods between the two fitted models from derivatives, but doing so would require either many derivative evaluations (which could then be numerically integrated along some path to obtain the difference in loglikelihoods), or we could try using Taylor series approximations to the loglikelihood. In the present case, however, the estimated standard errors of the parameters are sufficiently small that there is no doubt the model with separate parameters for each region fits better than the model with common parameters. For example, based on asymptotic normality, the approximate 95% confidence intervals for α in the circle and rectangle are, respectively, $(2.091, 2.172)$ and $(3.352, 3.518)$, indicating that a model with a common α is not tenable. Similarly, there is overwhelming evidence that adding a nugget effect improves the fit over the models without a nugget effect in both regions.

It is not clear why the fitted models are so different in the two regions, but it is presumably related to the different shapes of the dense bodies. It would be interesting to explore the details of the finite element method to try to understand why the rectangle might produce deviations from discrete harmonicity that have a smoother continuous component (larger $\widehat{\alpha}$) and a smaller nugget ($\widehat{\theta}_0\widehat{\theta}_1 = 3793$ for the rectangle and 4852 for the circle).

7. Discussion.

We have demonstrated how derivatives of the loglikelihood function for a Gaussian process model can be accurately and efficiently calculated in situations for which direct calculation of the loglikelihood itself would be much more difficult. Being able to calculate these derivatives enables us to find solutions to the score equations and to verify that these solutions are at least local maximizers of the likelihood. However, if the score equations had multiple solutions, then, assuming all the solutions could be found, it might not be so easy to determine which was the global maximizer. Furthermore, as we saw in the previous section, it is not straightforward to obtain likelihood ratio statistics when only derivatives of the loglikelihood are available.

Perhaps a more critical drawback of having only derivatives of the loglikelihood occurs when using a Bayesian approach to parameter estimation. The likelihood needs to be known only up to a multiplicative constant, so, in principle, knowing the gradient of the loglikelihood throughout the parameter space is sufficient for calculating the posterior distribution. However, it is not so clear how one might calculate an approximate posterior based on just gradient and perhaps Hessian values of the loglikelihood at some discrete set of parameter values. It is even less clear how one could implement an MCMC scheme based on just derivatives of the loglikelihood.

Despite this substantial drawback, we consider the development of likelihood methods for fitting Gaussian process models that are nearly $O(n)$ in time and, perhaps more importantly $O(n)$ in memory, to be essential for expanding the scope of application of these models. We believe that the present work provides a useful step in this direction.

APPENDIX A: PROOFS

Proof of Theorem 2.1. Since K is positive definite, it can be written in the form $S\Lambda S'$ with S orthogonal and Λ diagonal with elements $\lambda_1 \geq \ldots \geq \lambda_n > 0$. Then $Q^i := S'K_iS$ is symmetric,

$$\text{(A.1)} \quad \operatorname{tr}(W^iW^j) = \operatorname{tr}(S'K^{-1}SS'K_iSS'K^{-1}SS'K_jS) = \operatorname{tr}(\Lambda^{-1}Q^i\Lambda^{-1}Q^j)$$

and, similarly,

$$\text{(A.2)} \quad \operatorname{tr}\{W^i(W^j)'\} = \operatorname{tr}(\Lambda^{-1}Q^iQ^j\Lambda^{-1}).$$

For real $v_1, \ldots, v_p$,

$$\text{(A.3)} \quad \sum_{i,j=1}^{p} v_iv_j \sum_{k=1}^{n} W^i_{kk}W^j_{kk} = \sum_{k=1}^{n}\left\{\sum_{i=1}^{p} v_iW^i_{kk}\right\}^2 \geq 0.$$

Furthermore, by (A.1),

$$\text{(A.4)} \quad \sum_{i,j=1}^{p} v_iv_j\operatorname{tr}(W^iW^j) = \sum_{k,\ell=1}^{n} \frac{1}{\lambda_k\lambda_\ell}\left\{\sum_{i=1}^{p} v_iQ^i_{k,\ell}\right\}^2,$$

and, by (A.2),

$$\sum_{i,j=1}^{p} v_i v_j \operatorname{tr}\{W^i (W^j)'\} = \sum_{k,\ell=1}^{n} \frac{1}{\lambda_k^2} \left\{ \sum_{i=1}^{p} v_i Q_{k,\ell}^i \right\}^2. \tag{A.5}$$

Write $\gamma_{k\ell}$ for $\sum_{i=1}^p v_i Q_{k,\ell}^i$ and note that $\gamma_{k\ell} = \gamma_{\ell k}$. Consider finding an upper bound to

$$\frac{\sum_{i,j=1}^p v_i v_j \operatorname{tr}\{W^i (W^j)'\}}{\sum_{i,j=1}^p v_i v_j \operatorname{tr}(W^i W^j)} = \frac{\sum_{k=1}^n \frac{a_{kk}^2}{\lambda_k^2} + \sum_{k>\ell} a_{k\ell}^2 \left(\frac{1}{\lambda_k^2} + \frac{1}{\lambda_\ell^2} \right)}{\sum_{k=1}^n \frac{a_{kk}^2}{\lambda_k^2} + \sum_{k>\ell} \frac{2a_{k\ell}^2}{\lambda_k \lambda_\ell}}.$$

Think of maximizing this ratio as a function of the $a_{k\ell}^2$'s for fixed λ_k's. We then have a ratio of two positively weighted sums of the same positive scalars (the $a_{k\ell}^2$'s for $k \geq \ell$), so this ratio will be maximized if the only positive $a_{k\ell}^2$ values correspond to cases for which the ratio of the weights, here

$$\frac{\frac{1}{\lambda_k^2} + \frac{1}{\lambda_\ell^2}}{\frac{2}{\lambda_k \lambda_\ell}} = \frac{1 + \left(\frac{\lambda_k}{\lambda_\ell}\right)^2}{\frac{2\lambda_k}{\lambda_\ell}}, \tag{A.6}$$

is maximized. Since we are considering only $k \geq \ell$, $\frac{\lambda_k}{\lambda_\ell} \geq 1$ and $\frac{1+x^2}{2x}$ is increasing on $[1, \infty)$, so (A.6) is maximized when $k = n$ and $\ell = 1$, yielding

$$\frac{\sum_{i,j=1}^p v_i v_j \operatorname{tr}\{W^i (W^j)'\}}{\sum_{i,j=1}^p v_i v_j \operatorname{tr}(W^i W^j)} \leq \frac{\kappa(K)^2 + 1}{2\kappa(K)}.$$

The theorem follows by putting this result together with (2.1), (2.2), and (A.3). □

Proof of Theorem 3.1. Define β_{ia} to be the ath element of β_i and $X_{\ell a}$ the ath diagonal element of X_ℓ. Then note that for $k \neq \ell$ and $k' \neq \ell'$ and $a, b \in \{1, \ldots, N\}$,

$$\begin{aligned} &(U_{i,(k-1)N+a} U_{i,(\ell-1)N+b},\, U_{j,(k'-1)N+a'} U_{j,(\ell'-1)N+b'}) \\ &= (\beta_{ia}\beta_{ib} Y_{ik} X_{ka} Y_{i\ell} X_{\ell b},\, \beta_{ja'}\beta_{jb'} Y_{jk'} X_{k'a'} Y_{j\ell'} X_{\ell' b'}) \end{aligned}$$

have the same joint distribution as for independent U_j's. Specifically, the two components are independent symmetric Bernoulli random variables unless $i = j, a = a', b = b'$ and $k = k' \neq \ell = \ell'$ or $i = j, a = b', b = a'$ and $k = \ell' \neq \ell = k'$, in which case, they are the same symmetric Bernoulli random variable. Straightforward calculations yield (3.3). □

ACKNOWLEDGMENTS

We thank Jed Brown at Argonne National Laboratory and Dave May at ETH Zurich for their help with the Stokes flow data. The work of Stein was supported by the U.S. Department of Energy grant No. DE-SC0002557 and the work of Chen and Anitescu was supported by the U.S. Department of Energy through contract No. DE-AC02-06CH11357.

REFERENCES

Anitescu, M., Chen, J. and Wang, L. (2012). A Matrix-free Approach for Solving the Parametric Gaussian Process Maximum Likelihood Problem. *SIAM Journal on Scientific Computing* **34** A240-A262.

Avron, H. and Toledo, S. (2011). Randomized algorithms for estimating the trace of an implicit symmetric positive semi-definite matrix. *Journal of the ACM* **58** 8:1–8:34.

Barnes, J. E. and Hut, P. (1986). A hierarchical $O(N \log N)$ force-calculation algorithm. *Nature* **324** 446–449.

Bhapkar, V. P. (1972). On a measure of efficiency of an estimating equation. *Sankhā A* **34** 467–472.

Box, G. E. P., Hunter, J. S. and Hunter, W. G. (2005). *Statistics for Experimenters: Design, Innovation, and Discovery*, second ed. Hoboken, NJ: John Wiley & Sons.

Caragea, P. C. and Smith, R. L. (2007). Asymptotic properties of computationally efficient alternative estimators for a class of multivariate normal models. *Journal of Multivariate Analysis* **98** 1417–1440.

Chan, R. H. and Jin, X. Q. (2007). *An Introduction to Iterative Toeplitz Solvers*. SIAM.

Chen, J., Anitescu, M. and Saad, Y. (2011). Computing f(A)b via Least Squares Polynomial Approximations. *SIAM Journal on Scientific Computing* **33** 195.

Chilés, J. P. and Delfiner, P. (1999). *Geostatistics: Modeling Spatial Uncertainty.* Wiley-Interscience.

Cressie, N. and Johannesson, G. (2008). Fixed rank kriging for very large spatial data sets. *Journal of the Royal Statistical Society* **70** 209–226.

Dahlhaus, R. and Künsch, H. (1987). Edge effects and efficient parameter estimation for stationary random fields. *Biometrika* **74** 887–892.

Dohrmann, C. R. and Bochev, P. B. (2004). A stabilized finite element method for the Stokes problem based on polynomial pressure projections. *International Journal for Numerical Methods in Fluids* **46** 183–201.

Eidsvik, M., Finley, A. O., Banerjee, S. and Rue, H. (2012). Approximate Bayesian inference for large spatial datasets using predictive process models. *Computational Statistics and Data Analysis* **56** 1362–1380.

Forsythe, G. E., Malcolm, M. A. and Moler, C. B. (1976). *Computer Methods for Mathematical Computations*. Prentice-Hall.

Fuentes, M. (2007). Approximate likelihood for large irregularly spaced spatial data. *Journal of the American Statistical Association* **102** 321–331.

Furrer, R., Genton, M. G. and Nychka, D. (2006). Covariance tapering for interpolation of large spatial datasets. *Journal of Computational and Graphical Statistics* **15** 502–523.

Furuichi, M., May, D. A. and Tackley, P. J. (2011). Development of a Stokes flow solver robust to large viscosity jumps using a Schur complement approach with mixed precision arithmetic. *Journal of Computational Physics* **230** 8835–8851.

Glimm, J., Lee, Y., Ye, K. Q. and Sharp, D. H. (2003). Prediction using numerical simulations, A Bayesian framework for uncertainty quantification and its statistical challenge. In *Uncertainty Modeling and Analysis, 2003. ISUMA 2003. Fourth International Symposium on* 380–385. IEEE.

Greengard, L. and Rokhlin, V. (1987). A Fast Algorithm for Particle Simulations. *J. Comput. Phys.* **73** 325–348.

Guyon, X. (1982). Parameter estimation for a stationary process on a d-dimensional lattice. *Biometrika* **69** 95–105.

Heyde, C. C. (1997). *Quasi-Likelihood and Its Application*. New York: Springer.

Hutchinson, M. F. (1990). A stochastic estimator of the trace of the influence matrix for Laplacian smoothing splines. *Communications in Statistics – Simulations* **19** 433–450.

Kaufman, C., Schervish, M. and Nychka, D. (2008). Covariance tapering for likelihood-based estimation in large spatial datasets. *Journal of the American Statistical Association* **103** 1556–1569.

Lee, Y. H. (2005). Stochastic Error Analysis of Multiscale Flow Simulations: The Two-phase Oil Reservoir Problem PhD thesis, Stony Brook University.

May, D. A. and Moresi, L. (2008). Preconditioned iterative methods for Stokes flow problems arising in computational geodynamics. *Physics of the Earth and Planetary Interiors* **171** 33–47.

O'Leary, D. P. (1980). The block conjugate gradient algorithm and related methods. *Linear Algebra Appl.* **29** 293–322.

Saad, Y. (2003). *Iterative Methods for Sparse Linear Systems*, second ed. SIAM.

Stein, M. L. (1995). Fixed domain asymptotics for spatial periodograms. *Journal of the American Statistical Association* **90** 1277–1288.

Stein, M. L. (1999). *Interpolation of Spatial Data: Some Theory for Kriging*. New York: Springer.

Stein, M. L. (2004). Equivalence of Gaussian measures for some nonstationary random fields. *Journal of Statistical Planning and Inference* **123** 1–11.

Stein, M. L. (2007). A modeling approach for large spatial datasets. *Journal of the Korean Statistical Society* **37** 3–10.

Stein, M. L. (submitted). Statistical properties of covariance tapers. *Journal of Computational and Graphical Statistics*.

Stein, M. L., Chen, J. and Anitescu, M. (2012). Difference Filter Preconditioning for Large Covariance Matrices. *SIAM J. Matrix Anal. Appl.* **33** 52–72.

Stein, M. L., Chi, Z. and Welty, L. J. (2004). Approximating likelihoods for large spatial datasets. *Journal of the Royal Statistical Society* **Series B, 66** 275–296.

Vecchia, A. V. (1988). Estimation and model identification for continuous spatial processes. *Journal of the Royal Statistical Society* **Series B, 50** 297–312.

Wang, D. and Loh, W. L. (2011). On fixed-domain asymptotics and covariance tapering in Gaussian random field models. *Electronic Journal of Statistics* **5** 238–269.

Whittle, P. (1954). On stationary processes in the plan. *Biometrika* **41** 434–449.

Zhang, Y. (2006). Uniformly distributed seeds for randomized trace estimator on $O(N^2)$-operation log-det approximation in Gaussian process regression. In *ICNSC '06. Proceedings of the 2006 IEEE International Conference on Networking, Sensing and Control* 498–503.

Argonne National Laboratory, Mathematics and Computer Science Division, Preprint ANL/MCS-P2091-0512

Department of Statistics
University of Chicago
Chicago, IL 60637
E-mail: stein@galton.uchicago.edu

Mathematics and Computer Science Division
Argonne National Laboratory
Argonne, IL 60439
E-mail: jiechen@mcs.anl.gov
anitescu@mcs.anl.gov